Turing AI 万维图灵 | 大数据系列课程

大数据技术-第九章：Sqoop数据迁移

Sqoop功能应用

CONTENTS

01

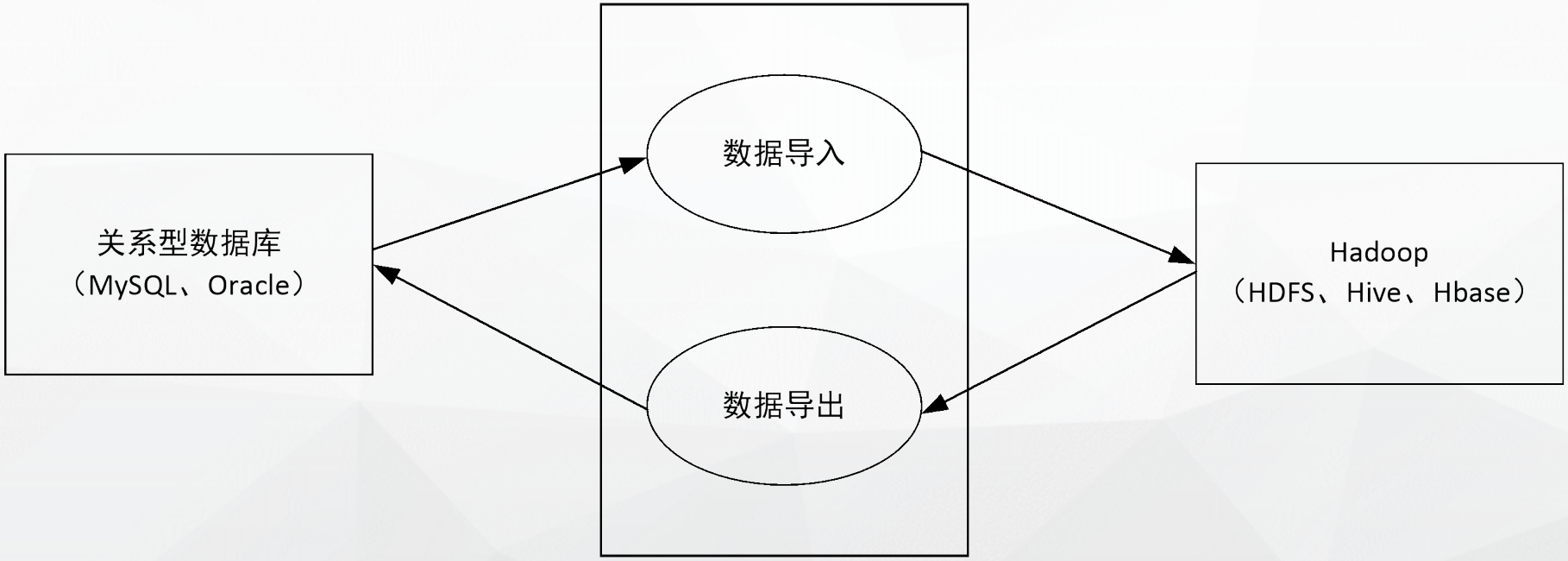
Sqoop架构

Sqoop架构

Sqoop是连接关系型数据库和Hadoop的桥梁，主要有两个方面(导入和导出)：

(1)将关系型数据库的数据导入到Hadoop 及其相关的系统中，如Hive和HBase。

(2)将数据从Hadoop系统里抽取并导出到关系型数据库。



Sqoop架构

Sqoop可以高效、可控的利用资源，可以通过调整任务数来控制任务的并发度。可以自动的完成数据映射和转换。由于导入数据库是有类型的，它可以自动根据数据库中的类型转换到Hadoop中，当然用户也可以自定义它们之间的映射关系。Sqoop支持多种数据库，如MySQL，Orcale等数据库。

Sqoop将导入或导出命令翻译成MapReduce程序来实现在翻译出的,MapReduce中主要是对InputFormat和OutputFormat进行定制。Sqoop命令使用Sqoop客户端直接提交代码，使用CLI命令行控制台方式访问，在命令或者脚本指定用户数据库名和密码。

Sqoop工具接收到客户端的shell命令或者Java API命令后，通过Sqoop中的任务翻译器(Task Translator)将命令转换为对应的MapReduce任务，再将关系型数据库和Hadoop中的数据进行相互转移，进而完成数据的拷贝。

Sqoop架构部署简单、使用方便，但也存在一些缺点，例如命令行方式容易出错，格式紧耦合，无法支持所有数据类型，安全机制不够完善，例如密码暴漏，安装需要root权限，connector必须符合JDBC模型。

02

Sqoop导入原理

Sqoop导入原理

在导入开始之前，Sqoop使用JDBC来检查将要导入的表。他检索出表中所有的列以及列的SQL数据类型。这些SQL类型（VARCHAR、INTEGER）被映射到Java数据类型（String、Integer等），在MapReduce应用中将使用这些对应的Java类型来保存字段的值。Sqoop的代码生成器使用这些信息来创建对应表的类，用于保存从表中抽取的记录。对于导入来说，更关键的是DBWritable接口的序列化方法，这些方法能使Widget类和JDBC进行交互：

```
Public void readFields(resultSet _dbResults)throws SQLException;

Public void write(PreparedStatement _dbstmt)throws SQLException;
```

Sqoop导入原理

JDBC的ResultSet接口提供了一个用户从检查结果中检索记录的游标；这里的ReadFields()方法将用ResultSet中一行数据的列来填充Example对象的字段。Sqoop启动的MapReduce作业用到一个InputFormat,他可以通过JDBC从一个数据库表中读取部分内容。Hadoop提供的DataDriverDBInputFormat能够为几个Map任务对查询结果进行划分。为了获取更好的导入性能，查询会根据一个“划分列”来进行划分的。Sqoop会选择一个合适的列作为划分列（通常是表的主键）。在生成反序列化代码和配置InputFormat之后，Sqoop将作业发送到MapReduce集群。Map任务将执行查询并将ResultSet中的数据反序列化到生成类的实例，这些数据要么直接保存在SequenceFile文件中，要么在写到HDFS之前被转换成分割的文本。Sqoop不需要每次都导入整张表，用户也可以在查询中加入到where子句，以此来限定需要导入的记录：Sqoop –query <SQL>。在向HDFS导入数据时，重要的是要确保访问的是数据源的一致性快照。从一个数据库中并行读取数据的Map任务分别运行在不同的进程中。因此，他们不能共享一个数据库任务。保证一致性的最好方法就是在导入时不允许运行任何进行对表中现有数据进行更新。

03

Sqoop导出原理

Sqoop导出原理

Sqoop导出功能的架构与其导入功能非常相似，在执行导出操作之前，Sqoop会根据数据库连接字符串来选择一个导出方法。一般为JDBC。然后，Sqoop会根据目标表的定义生成一个Java类。这个生成的类能够从文本文件中解析记录，并能够向表中插入类型合适的值。接着会启动一个MapReduce作业，从HDFS中读取源数据文件，使用生成的类解析记录，并且执行选定的导出方法。

Sqoop导出原理

基于JDBC的导出方法会产生一批insert语句，每条语句都会向目标表中插入多条记录。多个单独的线程被用于从HDFS读取数据并与数据库进行通信，以确保涉及不同系统的I/O操作能够尽可能重叠执行。虽然HDFS读取数据的MapReduce作业大多根据所处理文件的数量和大小来选择并行度（Map任务的数量），但Sqoop的导出工具允许用户明确设定任务的数量。由于导出性能会受并行的数据库写入线程数量的影响，所以Sqoop使用combinefileinput类将输入文件分组分配给少数几个map任务去执行。进程的并行特性导致导出操作往往不是原子操作。Sqoop会采用多个并行的任务导出，并且数据库系统使用固定大小的缓冲区来存储事务数据，这时一个任务中的所有操作不可能在一个事务中完成。因此，在导出操作进行过程中，提交过的中间结果都是可见的。在导出过程完成前，不要启动那些使用导出结果的应用程序，否则这些应用会看到不完整的导出结果。更有问题的是，如果任务失败，该任务将从头开始重新导入自己负责的那部分数据，因此可能会插入重复的记录。当前Sqoop还不能避免这种可能性。在启动导出作业前，应当在数据库中设置表的约束（例如，定义一个主键列）以保证数据行的唯一性。

Turing AI 万维图灵 | 大数据系列课程

大数据

BIG DATA

智 / 能 / 科 / 技　　放 / 眼 / 未 / 来